

Nyström Methods in the RKQ Algorithm for Initial-value Problems

J.S.C. Prentice
 Department of Applied Mathematics
 University of Johannesburg
 South Africa

February 7, 2012

Abstract

We incorporate explicit Nyström methods into the RKQ algorithm for stepwise global error control in numerical solutions of initial-value problems. The initial-value problem is transformed into an explicitly second-order problem, so as to be suitable for Nyström integration. The Nyström methods used are fourth-order, fifth-order and 10th-order. Two examples demonstrate the effectiveness of the algorithm.

1 Introduction

In two previous papers we have considered the *RKrvQz* algorithm for stepwise control of the global error in the numerical solution of an initial-value problem (IVP), using Runge-Kutta methods [1][2]. In the current paper, the third in the series, we focus our attention on the use of Nyström methods in this error control algorithm for n -dimensional problems of the form

$$\begin{aligned} \mathbf{y}''(x) &= \mathbf{f}(x, \mathbf{y}) \\ \mathbf{y}(x_0) &= \mathbf{y}_0 \\ \mathbf{y}'(x_0) &= \mathbf{y}'_0. \end{aligned} \tag{1}$$

Note that \mathbf{f} is not dependent on \mathbf{y}' . We designate this Nyström-based algorithm *RKNrvQz*, and we will show in a later section how any first-order IVP can be written in the form (1), so that *RKNrvQz* is, in fact, generally applicable. The motivation for considering this modification to *RKrvQz*

is twofold: most physical systems are described by second-order differential equations, and Nyström methods applied to (1) tend to be more efficient than their Runge-Kutta counterparts.

2 Relevant Concepts, Terminology and Notation

Here we describe concepts, terminology and notation relevant to our work. Note that boldface quantities are $n \times 1$ vectors, except for $\boldsymbol{\alpha}_i^r$, \mathbf{I}_n , \mathbf{F}_y^r , $\mathbf{F}_{y'}^r$ and \mathbf{g}_y , which are $n \times n$ matrices.

2.1 Nyström Methods

The most general definition of a Nyström method (sometimes known as Runge-Kutta-Nyström (RKN)) for solving (1) is

$$\begin{aligned} \mathbf{k}_p &= \mathbf{f} \left(x_i + c_p h_i, \mathbf{w}_i + c_p h_i \mathbf{w}'_i + h_i^2 \sum_{q=1}^m a_{pq} \mathbf{k}_q \right) \quad p = 1, 2, \dots, m \\ \mathbf{w}_{i+1} &= \mathbf{w}_i + h_i \mathbf{w}'_i + h_i^2 \sum_{p=1}^m b_p \mathbf{k}_p \equiv \mathbf{w}_i + h_i \mathbf{F}(x_i, \mathbf{w}_i) \\ \mathbf{w}'_{i+1} &= \mathbf{w}'_i + h_i \sum_{p=1}^m \hat{b}_p \mathbf{k}_p. \end{aligned} \quad (2)$$

The coefficients c_p , a_{pq} , b_p and \hat{b}_p are unique to the given method. If $a_{pq} = 0$ for all $p \leq q$, then the method is said to be *explicit*; otherwise, it is known as an *implicit* RKN method. We will focus our attention on explicit methods. In the second line of (2), we have implicitly defined the function \mathbf{F} . We treat \mathbf{w}'_i as an ‘internal parameter’; for our purposes here, we do not identify \mathbf{w}' with \mathbf{y}' , because \mathbf{f} is not dependent on \mathbf{y}' . The symbol \mathbf{w} is used here and throughout to indicate the approximate numerical solution, whereas the symbol \mathbf{y} will be used to denote the exact solution. We will denote an RKN method of order r as RKN r and, for such a method, we write

$$\mathbf{w}_{i+1}^r = \mathbf{w}_i^r + h_i \mathbf{F}^r(x_i, \mathbf{w}_i^r, \mathbf{w}_i^{r'}) . \quad (3)$$

The stepsize h_i is given by

$$h_i \equiv x_{i+1} - x_i$$

and carries the subscript because it may vary from step to step. It is known that RKN r has a local error of order $r + 1$ and a global error of order r , just like its Runge-Kutta counterpart RK r .

2.2 IVPs in the form $y'' = f(x, \mathbf{y})$

Consider the n -dimensional IVP

$$\begin{aligned}\mathbf{y}'(x) &= \mathbf{g}(x, \mathbf{y}) \\ \mathbf{y}(x_0) &= \mathbf{y}_0.\end{aligned}\tag{4}$$

This gives

$$y_j'' = \sum_{i=1}^n \frac{\partial g_j(x, \mathbf{y})}{\partial y_i} \frac{dy_i}{dx} = \sum_{i=1}^n \frac{\partial g_j(x, \mathbf{y})}{\partial y_i} g_i(x, \mathbf{y})$$

where y_i is the i th component of \mathbf{y} , and g_i is the i th component of \mathbf{g} . Clearly, we have

$$y_j'' = \sum_{i=1}^n \frac{\partial g_j(x, \mathbf{y})}{\partial y_i} g_i(x, \mathbf{y}) \equiv f_j(x, \mathbf{y})$$

for all $j = 1, 2, \dots, n$, and so we can write

$$\mathbf{y}''(x) = \mathbf{f}(x, \mathbf{y}).$$

The initial values for this second-order problem are then given by

$$\begin{aligned}\mathbf{y}(x_0) &= \mathbf{y}_0 \\ \mathbf{y}'(x_0) &= \mathbf{g}(x_0, \mathbf{y}_0) \equiv \mathbf{y}_0' .\end{aligned}$$

Hence, any first-order IVP can be transformed into an IVP of the form (1). This is ideally suited to the Nyström methods, which are specifically designed for this type of IVP. They are also more efficient than their Runge-Kutta counterparts; for example, the methods to be used later, RKN4 and RKN5, require three and four stage evaluations, respectively, as opposed to RK4 and RK5, which require at least four and six stage evaluations, respectively.

2.3 Error Propagation in RKN

It can be shown [3] that, for $\text{RKN}r$,

$$\Delta_{i+1}^r \equiv \mathbf{w}_{i+1}^r - \mathbf{y}_{i+1} = \boldsymbol{\varepsilon}_{i+1}^r + \boldsymbol{\alpha}_i^r \Delta_i^r \tag{5}$$

$$\boldsymbol{\alpha}_i^r \equiv \mathbf{I}_n + h_i \mathbf{F}_y^r(x_i, \boldsymbol{\xi}_i), \tag{6}$$

where $\boldsymbol{\varepsilon}_{i+1}^r = O(h_i^{r+1})$ is the local error, Δ_{i+1}^r is the global error and \mathbf{F}_y^r is the Jacobian (with respect to \mathbf{y}) of the function $\mathbf{F}^r(x_i, \mathbf{w}_i^r)$ associated with $\text{RKN}r$. The term $h_i \mathbf{F}_y^r(x_i, \boldsymbol{\xi}_i)$ in the matrix $\boldsymbol{\alpha}_i^r$ arises from a first-order Taylor expansion of $\mathbf{F}^r(x_i, \mathbf{w}_i) = \mathbf{F}^r(x_i, \mathbf{y}_i + \Delta_i^r)$ with respect to \mathbf{y}_i .

For a Nyström method RKN r , we have $\mathbf{F}^r = \mathbf{F}^r(x_i, \mathbf{w}_i^r)$ and so, as above,

$$\boldsymbol{\alpha}_i^r \equiv \mathbf{I}_n + h_i \mathbf{F}_y^r(x_i, \boldsymbol{\zeta}_i),$$

where $\boldsymbol{\zeta}_i$ is an appropriate constant. Hence, the global error in RKN r is also given by (5).

2.4 RK rv Q z

We will not discuss RK rv Q z in detail here; the reader is referred to our previous work where the algorithm has been discussed extensively. It suffices to say that RK rv Q z uses RK r and RK v to control local error via local extrapolation, while simultaneously using RK z to keep track of the global error in the RK r solution. Such global error arises due to the propagation of the RK v global error. RK rv Q z is designed to estimate the various components of the global error in RK r and RK v at each node and, when the global error is deemed too large, a quenching procedure is carried out. This simply involves replacing the RK r and RK v solutions with the much more accurate RK z solution, whenever necessary, so that the RK r and RK v global errors do not accumulate beyond a desired tolerance.

2.5 RKN rv Q z

The algorithm RKN rv Q z is nothing more than RK rv Q z with RK r , RK v and RK z replaced with RKN r , RKN v and RKN z . Of course, RKN rv Q z is applied to problems of the form (1), whereas RK rv Q z is applied to problems of the form (4).

We also report on a refinement to the algorithm: in RK rv Q z , if the global error at x_i is too large, we replace \mathbf{w}_i^r with \mathbf{w}_i^z and then recompute \mathbf{w}_{i+1}^r and \mathbf{w}_{i+1}^v , using \mathbf{w}_i^z as input for both RK r and RK v . This is the essence of the quenching procedure. However, in retrospect it seems quite acceptable to simply replace \mathbf{w}_{i+1}^r and \mathbf{w}_{i+1}^v with \mathbf{w}_{i+1}^z ; this avoids the need for recomputing \mathbf{w}_{i+1}^r and \mathbf{w}_{i+1}^v , which improves efficiency and, after all, it is the global error in \mathbf{w}_{i+1}^r and \mathbf{w}_{i+1}^v , not \mathbf{w}_i^r and \mathbf{w}_i^v , that is too large. Both approaches are effective, although one is more efficient than the other. It is the more efficient approach that we have employed in RKN rv Q z .

3 Numerical Examples

It is not our intention to compare methods or algorithms but, for the sake of consistency, we will apply RKN rv Q z to the same examples that we considered in our previous work on RK rv Q z . In our calculations, we use RKN4,

RKN5 and RKN10 which gives the algorithm RKN45Q10. RKN4 and RKN5 are taken from Hairer et al [5], and RKN10 is from Dormand et al [4].

The first of these is the scalar problem

$$\begin{aligned} y' &= \left(\frac{\ln 1000}{100} \right) y \\ y(0) &= 1 \end{aligned}$$

which transforms to

$$\begin{aligned} y'' &= \left(\frac{\ln 1000}{100} \right)^2 y \\ y(0) &= 1 \\ y'(0) &= \frac{\ln 1000}{100}. \end{aligned}$$

Solving this problem with RKN45 and RKN45Q10 with a tolerance of 10^{-10} on the absolute local and global errors gives the error curves shown in Figure 1. The global error obtained with RKN45 is clearly larger than the desired tolerance on most of the interval, despite local error control via local extrapolation. However, RKN45Q10 yields a solution with a global error always less than the tolerance - the maximum global error in this case is 9.1×10^{-11} . The points on the x -axis where this global error decreases sharply correspond to the quenches carried out using RKN10.

The second example is the simple harmonic oscillator

$$\begin{aligned} y_1' &= y_2 \\ y_2' &= -y_1 \\ \mathbf{y}(0) &= \begin{bmatrix} 0 \\ 1000 \end{bmatrix} \end{aligned}$$

which has solution

$$\begin{aligned} y_1(x) &= 1000 \sin x \\ y_2(x) &= 1000 \cos x \end{aligned}$$

and becomes, in explicit second-order form,

$$\begin{aligned} \mathbf{y}'' &= \begin{bmatrix} y_1'' \\ y_2'' \end{bmatrix} = \begin{bmatrix} -y_1 \\ -y_2 \end{bmatrix} \equiv \mathbf{f}(x, \mathbf{y}) \\ \mathbf{y}(0) &= \begin{bmatrix} 0 \\ 1000 \end{bmatrix}, \mathbf{y}'(0) = \begin{bmatrix} 1000 \\ 0 \end{bmatrix}. \end{aligned}$$

Since the solution oscillates between -1000 and 1000 , there are regions where the solution has magnitude less than unity - here, we implement absolute error control - and regions where the solution has magnitude greater than unity, where we implement relative error control. With an imposed tolerance of 10^{-8} on the local and global errors (relative and absolute) we found a maximum global error of $\sim 4 \times 10^{-8}$ in each component when using RKN45, and a global error no greater than 0.99×10^{-8} with RKN45Q10, on $x \in [0, 200]$. A total of 20 quenches were needed.

4 Conclusion

We have considered the use of Nyström methods in RKrvQz, wherein a combination of local extrapolation and quenching result in stepwise global error control in numerical solutions of IVPs. Two examples have demonstrated the success of RKN45Q10.

References

- [1] Prentice, J.S.C. (2011). Stepwise Global Error Control in an Explicit Runge-Kutta Method using Local Extrapolation with High-Order Selective Quenching, *Journal of Mathematics Research*, 3, 2, 126-136. [<http://ccsenet.org/journal/index.php/jmr/article/view/8700/7481>]
- [2] Prentice, J.S.C. (2011). Relative Global Error Control in the RKQ Method for Systems of Ordinary Differential Equations, *Journal of Mathematics Research*, 3, 4, 59-66. [<http://ccsenet.org/journal/index.php/jmr/article/view/10491/8952>]
- [3] Prentice, J.S.C. (2009). General error propagation in the RKrGLm method, *Journal of Computational and Applied Mathematics*, 228, 344-354.
- [4] Dormand, J.R., El-Mikkawy, M.E.A., and Prince, J. (1987). High-Order Embedded Runge-Kutta-Nyström Formulae, *IMA Journal of Numerical Analysis*, 7, 423-430.
- [5] Hairer, E., Norsett, S.P., and Wanner, G. (2000). *Solving Ordinary Differential Equations I: Nonstiff Problems*, Berlin: Springer

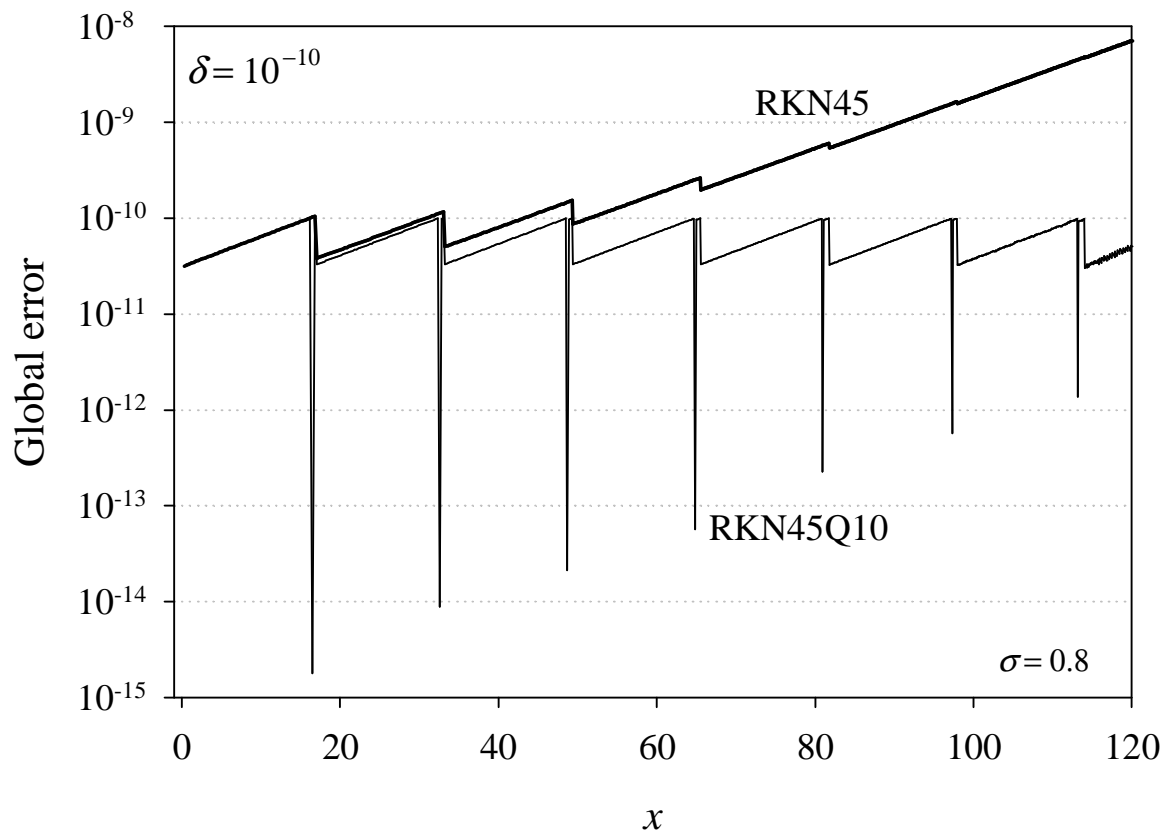


Figure 1: Global errors for RKN45 and RKN45Q10 applied to the exponential test problem.